

WHAT IS CLAIMED IS:

1 1. A method for use in a fixed point arithmetic processing device having
2 an input vector that contains one or more vector elements, and is an M bit integer, and a
3 maximum permitted left shift (MLS) value for the input vector is less than or equal to the
4 value of M - 2, the method for scaling all the vector elements based on the vector element
5 with the largest magnitude, the method comprising:
6 sequentially searching each vector element to find a left shift value for scaling
7 each vector element;
8 comparing the left shift values to determine a minimum left shift (NLS_MIN)
9 for scaling the largest vector element;
10 employing the NLS_MIN value to determine whether the input vector is a
11 non-zero input vector;
12 if so, regardless of whether the largest element of non-zero input vector has a
13 positive or negative magnitude, offsetting the NLS_MIN value by the MLS value to obtain
14 an actual number of left shifts (NLS) value for scaling the input vector;
15 determining whether the input vector is a zero input vector; and
16 if so, offsetting the NLS_MIN value by the MLS value to obtain the NLS
17 value.

18 2. The method of claim 1 further comprising employing a pdmsb
19 instruction for sequentially searching, and for comparing said left shift values.

20 3. A method, by a processing device, for scaling an M-bit integer input
21 vector containing one or more vector elements, said method comprising:
22 receiving a maximum permitted left shift (MLS) value for the input vector,
23 said MLS value being less than or equal to M - 2;
24 determining a minimum left shift (NLS_MIN) for scaling said vector element
25 with the largest magnitude;
26 employing said NLS_MIN value to determine whether said input vector is a
27 zero input vector, or a non-zero input vector irrespective of the positive or negative value of
28 said non-zero input vector;
29 if a non-zero input vector is determined, offsetting said NLS_MIN value by
30 said MLS value to obtain an actual number of left shifts (NLS) value for scaling said input
31 vector; and
32

13 if a zero input vector is determined, offsetting said NLS_MIN value by said
14 MLS value to obtain the NLS value.

1 4. The method of claim 3 wherein offsetting said NLS_MIN value for
2 said zero input vector further comprises said NLS value being given by: $MLS + 1$.

1 5. The method of claim 3 wherein offsetting said NLS_MIN value for
2 said non-zero input vector further comprises said NLS value given by: $NLS = NLS_MIN +$
3 $(MLS - (M - 2))$.

1 6. The method of claim 3 further comprising employing a pdmsb
2 instruction for sequentially searching, and for comparing said left shift values.

1 7.. The method of claim 3 wherein employing said NLS_MIN value
2 further comprises determining whether $NLS_MIN = 31$, if $NLS_MIN \neq 31$, then the input
3 vector is a non-zero input vector.

1 8. A processor operable from an M-bit instruction set where M is an
2 integer, the processor comprising:

3 a memory unit for storing at least first instruction stream comprising M-bit
4 instructions;

5 an execution unit operable to receive execution signals to execute the M-bit
6 instructions;

7 a decode unit coupled to the memory unit and to the execution unit to receive
8 and decode the first instruction stream from the memory unit to produce therefrom the
9 execution signals, the execution signals for:

10 determining a minimum left shift (NLS_MIN) for scaling said vector element
11 with the largest magnitude;

12 employing said NLS_MIN value to determine whether said input vector is a
13 zero input vector, or a non-zero input vector by evaluating if $NLS_MIN = 31$;

14 if $NLS_MIN \neq 31$, then the input vector is a non-zero input vector; and

15 determining an actual number of left shifts (NLS) for scaling the non-zero
16 input vector.

